Data-Driven MCMC for Learning and Inference in Switching Linear Dynamic Systems

Sang Min Oh James M. Rehg Tucker Balch Frank Dellaert
College of Computing, Georgia Institute of Technology
801 Atlantic Drive Atlanta, GA 30332-0280
{sangmin, tucker, rehg, dellaert}@cc.gatech.edu

Abstract

Switching Linear Dynamic System (SLDS) models are a popular technique for modeling complex nonlinear dynamic systems. An SLDS has significantly more descriptive power than an HMM, but inference in SLDS models is computationally intractable. This paper describes a novel inference algorithm for SLDS models based on the Data-Driven MCMC paradigm. We describe a new proposal distribution which substantially increases the convergence speed. Comparisons to standard deterministic approximation methods demonstrate the improved accuracy of our new approach. We apply our approach to the problem of learning an SLDS model of the bee dance. Honeybees communicate the location and distance to food sources through a dance that takes place within the hive. We learn SLDS model parameters from tracking data which is automatically extracted from video. We then demonstrate the ability to successfully segment novel bee dances into their constituent parts, effectively decoding the dance of the bees.

Introduction

Switching Linear Dynamic System (SLDS) models have been studied in a variety of problem domains. Representative examples include computer vision (North *et al.* 2000; Pavlović *et al.* 1999; Bregler 1997), computer graphics (Y.Li, T.Wang, & Shum 2002), speech recognition (Rosti & Gales 2004), econometrics (Kim 1994), machine learning (Lerner *et al.* 2000; Ghahramani & Hinton 1998), and statistics (Shumway & Stoffer 1992). While there are several versions of SLDS in the literature, this paper addresses the model structure depicted in Figure 3. An SLDS model represents the nonlinear dynamic behavior of a complex system by switching among a set of linear dynamic models over time. In contrast to HMM's, the Markov process in an SLDS selects from a set of continuously-evolving linear Gaussian dynamics, rather than a fixed Gaussian mixture density. As a consequence, an SLDS has potentially greater descriptive power. Offsetting this advantage is the fact that exact inference in an SLDS is intractable, which complicates estimation and parameter learning (Lerner & Parr 2001).

Inference in an SLDS model involves computing the posterior distribution of the hidden states, which consist of the (discrete) switching state and the (continuous) dynamic state. In the behavior recognition application which motivates this work, the discrete state represents distinct honeybee behaviors while the dynamic state represents the bee's true motion. Given video-based measurements of the position and orientation of the bee over time, SLDS inference can be used to obtain a MAP estimate of the behavior and motion of the bee. In addition to its central role in applications such as MAP estimation, inference is also the crucial step in parameter learning via the EM algorithm (Pavlović, Rehg, & MacCormick 2000). Approximate inference techniques have been developed to address the computational limitations of the exact approach.

Previous work on approximate inference in SLDS models has focused primarily on two classes of techniques: stage-wise methods such as approximate Viterbi or GPB2 which maintain a constant representational size for each time step as data is processed sequentially, and structured variational methods which approximate the intractable exact model with a tractable, decoupled model (Pavlović, Rehg, & MacCormick 2000; Ghahramani & Hinton 1998). While these approaches are successful in some application domains, such as vision and graphics, they do not provide any mechanism for fine-grained control over the accuracy of the approximation. In fields such as biology where learned models can be used to answer scientific questions about animal behavior, scientists would like to characterize the accuracy of an approximation and they may be willing to pay an additional computational price for getting as close as possible to the true posterior.

In these situations where a controllable degree of accuracy is required to support a diverse range of tasks, Markov-Chain Monte-Carlo (MCMC) methods are attractive. Standard MCMC techniques, however, are often plagued by extremely slow convergence rates. We therefore explore the use of Rao-Blackwellisation (Casella & Robert 1996) and the Data-Driven MCMC paradigm (Tu & Zhu 2002) to improve convergence. We describe a novel proposal distribution for Data-driven MCMC SLDS inference and demonstrate its effectiveness in learning models from noisy real-world data. We believe this paper is the first to explore Data-Driven MCMC techniques for SLDS models.

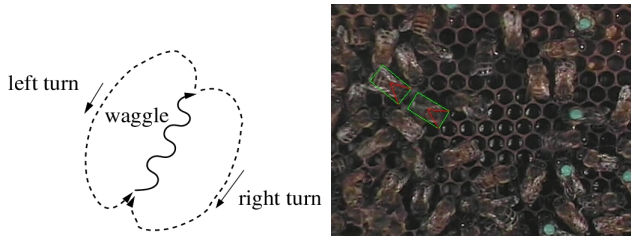

Fig. 1. (Left) A bee dance is in three patterns : *waggle, left turn, and right turn.* (Right) Green boxes are tracked bees.

Moreover, we believe that these techniques are broadly-applicable to problems in time series and behavior modeling.

The application domain which motivates this work is a new research area which enlists visual tracking and AI modeling techniques in the service of biology (Balch, Khan, & Veloso 2001). The current state of biological field work is still dominated by manual data interpretation, a time-consuming and error-prone process. Automatic interpretation methods can provide field biologists with new tools for quantitative studies of animal behavior. A classical example of animal behavior and communication is the honeybee dance, depicted in a stylized form in Fig. 1. Honeybees communicate the location and distance to food sources through a dance that takes place within the hive. The dance is decomposed into three different regimes: "turn left", "turn right" and "waggle".

Previously, we developed a vision system that automatically tracks the dancer bees, which are shown with green rectangles in Fig.1. Tracks from multiple bee dances, extracted automatically by our system, are shown in Fig.7, where the different dance phases are illustrated in different colors. We apply the Data-Driven MCMC techniques that we have developed to this bee dance data and demonstrate successful modeling and segmentation of dance regimes. These experimental results demonstrate the effectiveness of our new framework.

Related Work

Switching linear dynamic system (SLDS) models have been studied in a variety of research communities ranging from computer vision (Pavlović *et al.* 1999; North *et al.* 2000; Bregler 1997; Soatto, Doretto, & Wu 2001), computer graphics (Y.Li, T.Wang, & Shum 2002), and speech recognition (Rosti & Gales 2004) to econometrics (Kim 1994), machine learning (Lerner *et al.* 2000; Ghahramani & Hinton 1998), and statistics (Shumway & Stoffer 1992).

Exact inference in SLDS is intractable (Lerner & Parr 2001). Thus, there have been research efforts to derive efficient but approximate schemes. The early examples include GPB2 (Bar-Shalom & Li 1993), and Kalman filtering (Bregler 1997). More recent examples include a variational approximation (Pavlović, Rehg, & MacCormick 2000), expectation propagation (Zoeter & Heskes 2003), sequential Monte Carlo methods (Doucet, Gordon, & Krishnamurthy 2001) and Gibbs sampling (Rosti & Gales 2004).

The Data-Driven MCMC approach has been successfully applied in computer vision, e.g., image segmentation/parsing (Tu & Zhu 2002) and human pose estimation in still images (Lee & Cohen 2004). In data-driven approach, the convergence of MCMC methods (Gilks, Richardson, & Spiegelhalter 1996) is accelerated by incorporating an efficient sample proposal which is driven by the "cues" present in the data.

SLDS Background

A switching linear dynamic systems (SLDS) model describes the dynamics of a complex physical process by the switching between a set of linear dynamic systems (LDS).

Linear Dynamic Systems

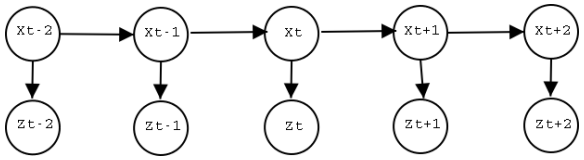

Fig. 2. A linear dynamic system (LDS)

An LDS is a time-series state-space model that comprises a linear Gaussian dynamics model and a linear Gaussian observation model. The graphical representation of an LDS is shown in Fig.2. The Markov chain at the top represents the state evolution of the continuous hidden states x_t. The prior density p_1 on the initial state x_1 is assumed to be normal with mean μ_1 and covariance Σ_1, i.e., $x_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$.

The state x_t is obtained by the product of state transition matrix F and the previous state x_{t-1} corrupted by the additive white noise w_t, zero-mean and normally distributed with covariance matrix Q:

$$x_t = Fx_{t-1} + w_t \text{ where } w_t \sim \mathcal{N}(0, Q) \qquad (1)$$

In addition, the measurement z_t is generated from the current state x_t through the observation matrix H, and corrupted by white observation noise v_t:

$$z_t = Hx_t + v_t \text{ where } v_t \sim \mathcal{N}(0, V) \qquad (2)$$

Thus, an LDS model M is defined by the tuple $M \triangleq \{(\mu_1, \Sigma_1), (F, Q), (H, V)\}$. Exact inference in an LDS can be done exactly using the RTS smoother (Bar-Shalom & Li 1993).

Switching Linear Dynamic Systems

An SLDS is a natural extension of an LDS, where we assume the existence of n distinct LDS models $M \triangleq \{M_i | 1 \leq i \leq n\}$, where each model M_i is defined by the LDS parameters. The graphical model corresponding to an SLDS is shown in Fig.3. The middle chain, representing the hidden state sequence $X \triangleq \{x_t | 1 \leq t \leq T\}$, together with the observations $Z \triangleq \{z_t | 1 \leq t \leq T\}$ at the bottom,

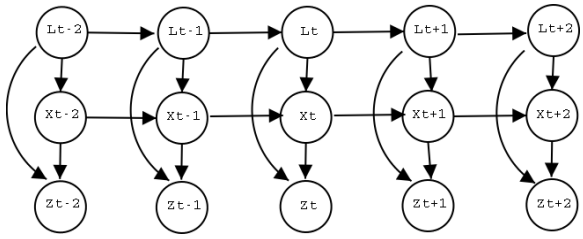

Fig. 3. Switching linear dynamic systems (SLDS)

is identical to an LDS in Fig.2. However, we now have an additional discrete Markov chain $L \triangleq \{l_t | 1 \leq t \leq T\}$ that determines which of the n models M_i is being used at every time-step. We call $l_t \in M$ the *label* at time t and L a *label sequence*.

In addition to a set of LDS models M, we specify two additional parameters: a multinomial distribution $\pi(l_1)$ over the initial label l_1 and an $n \times n$ transition matrix B that defines the switching behavior between the n distinct LDS models, i.e. $B_{ij} \triangleq P(l_j|l_i)$.

Learning in SLDS via EM

The EM algorithm (Dempster, Laird, & Rubin 1977) can be used to obtain the maximum-likelihood parameters $\hat{\Theta}$. The hidden variables in EM are the label sequence L and the state sequence X. Given the observation data Z, EM iterates between the two steps as in Algorithm 1.

Algorithm 1 EM for Learning in SLDS

- **E-step :** Inference to obtain the posterior distribution

$$f^i(L, X) \triangleq P(L, X|Z, \Theta^i) \tag{3}$$

  over the hidden variables L and X, using a current guess for the SLDS parameters Θ^i.
- **M-step :** maximize the expected log-likelihoods

$$\Theta^{i+1} \leftarrow \underset{\Theta}{\text{argmax}} \ \langle \log P(L, X, Z|\Theta) \rangle_{f^i(L,X)} \tag{4}$$

Above, $\langle \cdot \rangle_W$ denotes the expectation of a function $(\cdot)$ under a distribution W. The intractability of the exact E-step in Eq.3 motivates the development of approximate inference techniques.

Markov Chain Monte Carlo Inference

In this section, we introduce a novel sampling-based method that theoretically converges to the correct posterior distribution $P(L, X|Z, \Theta)$. Faster convergence is achieved by incorporating a data-driven approach where we introduce proposal priors and label-cue models.

Rao-Blackwellised MCMC

In our solution, we propose to pursue the Rao-Blackwellised posterior $P(L|Z, \Theta)$, rather than the joint posterior $P(L, X|Z, \Theta)$. The effect is the dramatic reduction of sampling space from L, X to L. This results in an improved approximation on the labels L, which are exactly the variables of interest in our application. This change is justified by the Rao-Blackwell theorem (Casella & Robert 1996). The Rao-Blackwellisation is achieved via the analytic integration on the continuous states X given a sample label sequence $L^{(r)}$. In this scheme, we can compute the probability of the r th sample labels $P(L^{(r)}|Z)$ up to a normalizing constant via the marginalization of the joint PDF :

$$P(L^{(r)}|Z) \quad = k \quad \int_X P(L^{(r)}, X, Z) \text{ where } k \triangleq \frac{1}{P(Z)} \tag{5}$$

Note that we omit the implicit dependence on the model parameters Θ for brevity. The joint PDF $P(L^{(j)}, X, Z)$ in the r.h.s. of Eq.5 can be obtained via the inference in the time-varying LDS with the varying but known parameters. Specifically, the inference over the continuous hidden states X in the middle chain of Fig.3 can be performed by RTS smoothing (Bar-Shalom & Li 1993). The resulting posterior is a series of Gaussians on X and can be effectively integrated out.

To generate samples from arbitrary distributions we can use the Metropolis-Hastings (MH) algorithm (Metropolis *et al.* 1953; Hastings 1970), a MCMC (Gilks, Richardson, & Spiegelhalter 1996). All MCMC methods work similarly : they generate a sequence of *samples* with the property that the collection of samples approximates the target distribution. To accomplish this, a *Markov chain* is defined over the space of labels L. The transition probabilities are set up in a very specific way such that the *stationary distribution* of the Markov chain is exactly the target distribution, in our case the posterior $P(L|Z)$. This guarantees that, if we run the chain for a sufficiently long time, the sample distribution converges to the target distribution.

Algorithm 2 Pseudo-code for Metropolis-Hastings (MH)

1) Start with a valid initial label sequence $L^{(1)}$.
2) Propose a new label sequence $L^{(r)'}$ from $L^{(r)}$ using a *proposal density* $Q(L^{(r)'}; L^{(r)})$.
3) Calculate the *acceptance ratio*

$$a = \frac{P(L^{(r)'}|Z)}{P(L^{(r)}|Z)} \frac{Q(L^{(r)}; L^{(r)'})}{Q(L^{(r)'}; L^{(r)})} \tag{6}$$

   where $P(L|Z)$ is the *target distribution*.
4) **If** $a >= 1$ then accept $L^{(r)'}$, i.e., $L^{(r+1)} \leftarrow L^{(r)'}$. **Otherwise**, accept $L^{(r)'}$ with probability $\min(1, a)$. If the proposal is rejected, then we keep the previous sample, i.e., $L^{(r+1)} \leftarrow L^{(r)}$.

We use the MH algorithm to generate a sequence of samples $L^{(r)}$'s until it converges. The pseudo-code for the MH algorithm is shown in Algorithm 2 (adapted from (Gilks, Richardson, & Spiegelhalter 1996)). Intuitively, step 2 proposes "moves" from the previous sample $L^{(r)}$ to the

next sample $L^{(r)'}$ in space L, which is driven by a proposal distribution $Q(L^{(r)'}; L^{(r)})$. The evaluation of a and the acceptance mechanism in steps 3 and 4 have the effect of modifying the transition probabilities of the chain in such a way that its stationary distribution is exactly $P(L|Z)$.

However, it is crucial to provide an efficient proposal Q, which results in faster convergence (Andrieu *et al.* 2003). Even if MCMC is guaranteed to converge, a naive exploration through the high dimensional state space L is prohibitive. Thus, the design of a proposal distribution which enhances the ability of the sampler to efficiently explore the space with high probability mass is motivated.

Temporal Cues and Proposal Priors

We propose to use *Data-Driven* paradigm (Tu & Zhu 2002) where the cues present in the data provide an efficient MCMC proposal distribution Q. Our data-driven approach is consisted of two phases : Learning and Inference.

In the *learning* phase, we first collect the temporal cues from the training data. Then, a set of label-cue models are constructed based on the collected cues. By a *temporal cue* c_t, we mean a cue at time t that can provide a guess for the corresponding label l_t. A cue c_t is a certain *statistic* obtained by observing the data within the fixed time range of z_t. We put a fixed-sized window on the data and obtain cues by looking inside it. Then, a set of n label-cue models $LC \triangleq \{P(c|l_i)|1 \leq i \leq n\}$ are learned from the classified cues where the cues are classified with respect to the training labels. Here, n corresponds to the number of existing patterns, the number of LDSs in our case. Each label-cue model $P(c|l_i)$ is an estimated generative model and describes the distribution of cue c given the label l_i.

In the *inference* phase, we first collect the temporal cues from the test data without access to the labels. Then, the learned label-cue models are applied to the cues and the proposal priors are constructed. A *proposal prior* $P(\tilde{l_t}|c_t)$ is a distribution on the labels, which is a rough approximation to the true posterior $P(l_t|Z)$. When a cue c_t is obtained from a test data, we construct a corresponding proposal prior $P(\tilde{l_t}|c_t)$ as follows :

$$P(\tilde{l_t}|c_t) \quad \triangleq \quad \frac{P(c_t|l_i)}{\sum_{i=1}^{n} P(c_t|l_i)} \tag{7}$$

In Eq.7, a proposal prior $P(\tilde{l_t}|c_t)$ is obtained from the normalized likelihoods of all labels. The prior describes the likelihood that each label generates the cue. By evaluating all the proposal priors across the test data, we obtain a full set of proposal priors $P(\tilde{L}) \triangleq \{P(\tilde{l_t}|c_t)|1 \leq t \leq T\}$ over the entire label sequence. However, the resulting proposal priors were found to be sensitive to the noise in the data. Thus, we smooth the estimates and use the resulting distribution. The proposed approach is depicted graphically in Fig.4,5,6 for the case of the bee dance domain.

In the honeybee dance, the change of heading angles is derived as a cue based on prior knowledge. From the stylized dance in Fig.1, we observe that the heading angles will jitter but stay constant on average during the waggling,

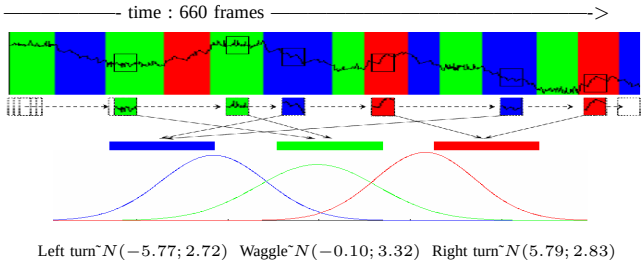

Fig. 4. Learning phase. Three label-cue models are learned from the training data. See text for detailed descriptions.

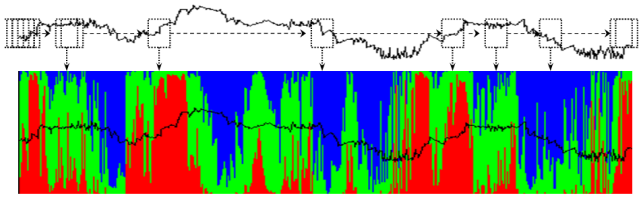

Fig. 5. Raw proposal priors. See text for descriptions.

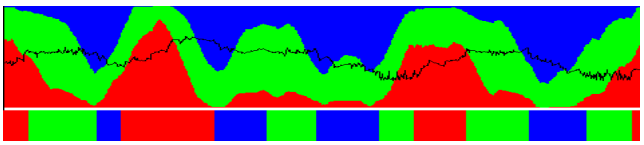

Fig. 6. Final proposal priors and the ground truth labels.

but generally increase or decrease during the right turn or left turn phases. Note that the heading angles are measured clockwise. Thus, a cue c_t for a frame is set to be the change of heading angles within the corresponding window.

In the learning phase, a cue window slides over the entire angle data while it collects cues as shown at the top of Fig.4. Then, the collected cues are classified according to the training labels. The label-cue(LC) models are estimated as the three Gaussians in our example, as shown at the bottom of Fig.4. The estimated means and the standard deviations show that the average change of heading angles are -5.77, -0.10 and 5.79 radians, as expected. In our example, the window size was set to 20 frames.

In the testing phase, the cues are collected from the data as shown at the top of Fig.5. Then, the proposal priors are evaluated based on the collected cues using Eq.7. The resulting label priors are shown in Fig.5. A slice of colors at t depicts the corresponding proposal prior $P(\tilde{l_t}|c_t)$. The raw estimates overfit the test data. Thus, we use the smoothed estimates as the final proposal priors, shown in Fig.6. At the bottom of Fig.6, the ground truth labels are shown below the final proposal priors for comparison. The obtained priors provide an excellent guide to the labels of the dance segments.

Data-Driven Proposal Distribution

The proposal priors $P(\tilde{L})$ and the SLDS discrete Markov transition PDF B constitute the data-driven proposal Q. The

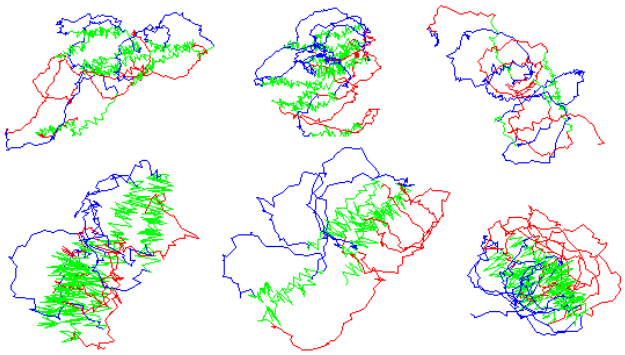

Fig. 7. Six dataset. 660, 660, 1054, 660, 609 and 814 frames

proposal scheme comprises two sub-procedures. First, it selects a local region to update based on the proposal priors. Rather than updating the entire sequence of a previous sample $L^{(r)}$, it selects a local region in $L^{(r)}$ and then proposes a locally updated new sample $L^{(r')}$. The local update scheme improves the space exploration capabilities of MCMC and results in faster convergence. Secondly, the proposal priors $P(\tilde{L})$ and the discrete transition PDF B are used to assign the new labels within a selected region. The second step has the effect of proposing a sample which reflects both the data and Markov properties of SLDSs. The two sub-steps are described in detail below.

In the first step, scoring schemes are used to select a local region within a sample. First, the previous sample labels $L^{(r)}$ are divided into a set of segments at a regular interval. Then, each segment is scored with respect to the proposal priors $P(\tilde{L})$, i.e. the affinities between the labels in each segment and the proposal priors are evaluated. Any reasonable affinity and scoring schemes are applicable. Finally, a segment is selected for an update via sampling based on the inverted scores.

In the second step, new labels l'_t's are sequentially assigned within a selected segment using the assignment function in Eq.8 where $B_{l'_t|l'_{t-1}} \triangleq P(l'_t|l'_{t-1})$. The implicit dependence of $\tilde{l}_t$ on c_t is omitted for brevity.

$$P(l'_t) = \beta\delta(l_t) + (1-\beta)\left\{\frac{B_{l'_t|l'_{t-1}}P(\tilde{l}'_t)}{\sum_{l'_t=1}^{n} B_{l'_t|l'_{t-1}}P(\tilde{l}'_t)}\right\} \quad (8)$$

Above, the first term with the sampling ratio β denotes the probability to keep the previous label l_t , i.e. $l'_t \leftarrow l_t$. The second term proposes a sampling of a new label l'_t. The Markov PDF B adds the model characteristics to a new sample. Consequently, the constructed proposal Q proposes samples that nicely embrace both the data and the intrinsic Markov properties. Then, MH algorithm balances the whole MCMC procedure in such a way that the MCMC inference on $P(L|Z)$ converges to the true posterior.

Experimental results

The experimental results on real-world bee data show the benefits of the new DD-MCMC framework. Six bee dance tracks in Fig.7 are obtained using a vision-based bee tracker. The observation data is a time-series sequence of four dimensional vectors $z_t = [x_t, y_t, \cos(\theta_t), \sin(\theta_t)]^T$where x_t,y_t and θ_t denote the 2D coordinates of bees and the heading angles. Note from Fig.7 that the tracks are noisy and much more irregular than the idealized stylized dance shown in Fig.1. The red, green and blue colors represent right-turn, waggle and left-turn phases. The ground-truth labels are marked manually for comparison and learning purposes. The dimensionality of the continuous hidden states was set to four.

Given the relative difficulty of obtaining accurate data, we adopted a leave-one-out strategy. The SLDS model and the label-cue models are learned from five out of six datasets, and the learned models are applied to the left-out dataset for automatic labeling. We interpreted the test dataset both by the proposed DD-MCMC method and the approximate Viterbi method for comparison. We selected the approximate Viterbi method as it known to be comparable to the other methods (Pavlović, Rehg, & MacCormick 2000). On average, 4,300 samples were used until the convergence for all the datasets.

Fig.8 shows an example posterior distribution $P(L|Z)$ which is discovered from the first dataset using the proposed DD-MCMC inference. The discovered posterior explicitly shows the uncertainties embedded in data. Multi-hypotheses are observed in the regions where the overlayed data shows irregular patterns. For example, around the two fifths from the right in Fig.8, the tracked bee systematically side-walks to the left due to the collision with other bees around it for about 20 frames while it was turning right. Consequently, the MCMC posterior shows the two eminent hypotheses for those frames : 70% turn left (blue) and 30% turn right (right) roughly.

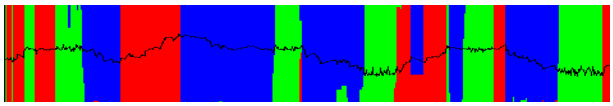

Fig. 8. Posterior distribution is discovered from Data 1.

The DD-MCMC is a Bayesian inference algorithm. Nonetheless, it can be used as a robust interpretation method. The MAP label sequence are taken from the discovered posterior distribution $P(L|Z)$. The resulting MCMC MAP labels, the ground-truth, and the approximate Viterbi labels for Data 1 are shown from the top to bottom in Fig. 9. It can be observed that DD-MCMC delivers solutions that concur very well with the ground truth. On the other hand, the approximate Viterbi labels at the bottom over-segments the data (insertion errors). The insertion errors of approximate Viterbi highlight one of the limitations of the class of deterministic algorithms for SLDS. In this respect, the proposed DD-MCMC inference method is shown to improve upon the Viterbi result and

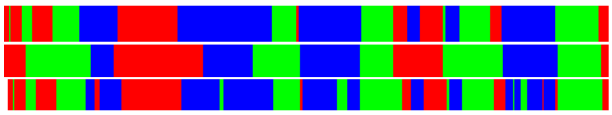

Fig. 9. Data 1 : DD-MCMC MAP, ground truth, Viterbi labels.

Errors	D 1	D 2	D 3	D 4	D 5	D 6
Insertions	7/14	5/6	2/5	3/4	1/1	0/0

TABLE I
MCMC MAP / VITERBI ERROR STATISTICS

provide more robust interpretation (inference) capabilities. The insertion errors induced by DD-MCMC MAP/Viterbi for all six datasets are summarized in Table.I.

Some errors between the MAP labels and the ground truth occur due to the systematic irregular motions of the tracked bees. In these cases, even an expert biologist will have difficulty figuring out all the correct dance labels soley based on the observation data, without access to the video. Considering that SLDSs are learned exclusively from the observation data, the results are fairly good.

Conclusion

We introduced a novel inference method for SLDS. The proposed method provides a concrete framework to a variety of research communities. First, it can characterize the accuracy of deterministic approximation algorithms. Secondly, it serves as an inference method that can discover true posteriors. This leads to a robust inference and learning. Finally, DD-MCMC delivers correct MAP solutions to a wide range of SLDS applications.

The proposed method efficiently converges to the posterior, overcoming a potential limitation of MCMC methods. The efficiency is achieved via the sampling space reduction by Rao-Blackwellisation and the data-driven approach where temporal cues and proposal priors are introduced to construct an efficient proposal.

In terms of characteristics, DD-MCMC inference method embraces data characteristics and model-based approaches. The characteristics of data are effectively learned using the temporal cues where the form of the cues are obtained from prior knowledge.

The authors believe that this work is the first to explore Data-Driven MCMC techniques for SLDS. Moreover, the proposed framework is broadly-applicable to problems in time series and behavior modeling.

References

Andrieu, C.; de Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to MCMC for machine learning. *Machine learning* 50:5–43.

Balch, T.; Khan, Z.; and Veloso, M. 2001. Automatically tracking and analyzing the behavior of live insect colonies. In *Proc. Autonomous Agents 2001*.

Bar-Shalom, Y., and Li, X. 1993. *Estimation and Tracking: principles, techniques and software*. Boston, London: Artech House.

Bregler, C. 1997. Learning and recognizing human dynamics in video sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Casella, G., and Robert, C. 1996. Rao-Blackwellisation of sampling schemes. *Biometrika* 83(1):81–94.

Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39(1):1–38.

Doucet, A.; Gordon, N. J.; and Krishnamurthy, V. 2001. Particle filters for state estimation of jump Markov linear systems. *IEEE Trans. Signal Processing* 49(3).

Ghahramani, Z., and Hinton, G. E. 1998. Variational learning for switching state-space models. *Neural Computation* 12(4):963–996.

Gilks, W.; Richardson, S.; and Spiegelhalter, D., eds. 1996. *Markov chain Monte Carlo in practice*. Chapman and Hall.

Hastings, W. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57:97–109.

Kim, C.-J. 1994. Dynamic linear models with Markov-switching. *Journal of Econometrics* 60.

Lee, M., and Cohen, I. 2004. Human upper body pose estimation in static images. In *Eur. Conf. on Computer Vision (ECCV)*.

Lerner, U., and Parr, R. 2001. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proc. 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, 310–318.

Lerner, U.; Parr, R.; Koller, D.; and Biswas, G. 2000. Bayesian fault detection and diagnosis in dynamic systems. In *Proc. AAAI*.

Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; and Teller, E. 1953. Equations of state calculations by fast computing machine. *Journal of Chemical Physics* 21:1087–1091.

North, B.; Blake, A.; Isard, M.; and Rottscher, J. 2000. Learning and classification of complex dynamics. *IEEE Trans. Pattern Anal. Machine Intell.* 22(9):1016–1034.

Pavlović, V.; Rehg, J.; Cham, T.-J.; and Murphy, K. 1999. A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Intl. Conf. on Computer Vision (ICCV)*.

Pavlović, V.; Rehg, J.; and MacCormick, J. 2000. Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems (NIPS)*.

Rosti, A.-V., and Gales, M. 2004. Rao-blackwellised Gibbs sampling for switching linear dynamical systems. In *Intl. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, volume 1, 809–812.

Shumway, R., and Stoffer, D. 1992. Dynamic linear models with switching. *Journal of the American Statistical Association* 86:763–769.

Soatto, S.; Doretto, G.; and Wu, Y. 2001. Dynamic Textures. In *Intl. Conf. on Computer Vision (ICCV)*, 439–446.

Tu, Z., and Zhu, S. 2002. Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Trans. Pattern Anal. Machine Intell.* 24(5):657–673.

Y.Li; T.Wang; and Shum, H.-Y. 2002. Motion texture : A two-level statistical model for character motion synthesis. In *SIGGRAPH*.

Zoeter, O., and Heskes, T. 2003. Hierarchical visualization of time-series data using switching linear dynamical systems. *IEEE Trans. Pattern Anal. Machine Intell.* 25(10):1202–1215.